This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) In a compiler, a method of determining a target type in an expression ~~with~~comprising an operator and at least one ~~undefined~~ expression operand in a loosely-typed programming language, comprising the steps of:

determining as said target type a most encompassed type from among a first set of types of loosely-typed operands, where said first set of types comprises all resulting types of all first variant expressions, where each of said first variant expressions comprises said target expression with at least one of said expression operands, at least one expression operand being a user-defined type~~wherein at least one operand is of an undefined type~~, replaced using widening type conversion, if said first set is not empty;

if said first set is empty, determining as said target type a most encompassing type from among a second set of types of loosely-typed operands, where said second set of types comprises all resulting types of all second variant expressions, where each of said second variant expressions comprises said target expression with at least one of said expression operands, ~~wherein at least one operand is of an undefined type~~, replaced using at least one of widening and narrowing type conversion; and

assigning said target type to said ~~undefined~~ operator.

2. (Previously presented) The method of claim 1, where said operands comprise n operands O_1 through O_n, where n is greater than or equal to one, and where each operand O_m is of a specific type T_m, where said step of determining as said target type said most encompassed type from among said first set of types comprises:

calculating said first set comprising types resulting from the operation of said operator on any possible set of hypothetical operands HO_1 through HO_n, where each hypothetical operand HO_m is of a type to which there is a widening conversion from type T_m ; and

if said first set is not empty, determining as said target type a most encompassed type from among said first set.

3. (Previously presented) The method of claim 2, wherein

if said first set is empty, calculating a second set comprising types resulting from the operation of said operator on any possible set of hypothetical operands HO_{n+1} through HO_{n+n}, where each hypothetical operand HO_{n+m} is of a type to which there is a conversion, either narrowing or widening, from type T_m; and

determining as said target type a most encompassing type from among said second set.

4. (Previously presented) The method of claim 2, said method further comprising:

converting each of said operands O_1 through O_n to said target type; and

computing said operation on said converted operands O_1 through O_n.

5. (Original) The method of claim 2 where said operator is a binary operator and n equals two.

6. (Original) The method of claim 2 where said operator is a unary operator and n equals one.

7. (Original) The method of claim 1 where said target type is an intrinsic type.

8. (Currently Amended) In a compiler, a method of resolving an expression comprising an overloaded binary operator, a first operand of a first type and a second operand of a second type, in a loosely-typed programming language, comprising the steps of:

parsing the expression to determine [[the]] said overloaded binary operator, the first operand and the second operand, the first operand and the second operand each being loosely typed, ~~wherein~~ at least one of the set comprising the first operand and the second operand being ~~is of an undefined~~ a user-defined type;

determining a first set of types, where said first set comprises all types to which there is a widening conversion from said first type;

determining a second set of types, where said second set comprises all types to which there is a widening conversion from said second type;

determining a third set of types, where said third set comprises all types which result from [[the]] an operation of said overloaded binary operator on a type from among said first set and a type from among said second set;

if said third set of types is empty, determining a fourth set of types, where said fourth set comprises all types to which there is a narrowing conversion from said first type and all types to which there is a widening conversion from said first type;

if said third set of types is empty, determining a fifth set of types, where said fifth set comprises all types to which there is a narrowing conversion from said second type and all types to which there is a widening conversion from said second type;

if said third set of types is empty, determining a sixth set of types, where said sixth set comprises all types which result from the operation of said overloaded binary operator on a type from among said fourth set and a type from among said fifth set;

if said third set of types is not empty, selecting the most encompassed type in said third set as a target type;

if said third set of types is empty, selecting the most encompassing type in said sixth set of types as said target type; and

assigning said target type to said ~~undefined~~ overloaded binary operator.

9. (Currently Amended) The method of claim 8, said method further comprising:

converting said first operand to said target type;

converting said second operand to said target type; and

computing said operation of said overloaded binary operator on said converted first operand and said converted second operand.

10. (Currently Amended) A computer-readable storage medium containing computer executable instructions for a compiler to resolve a target expression comprising an operator and at least one expression operand in a loosely-typed programming language, the computer-executable instructions to perform acts comprising:

determining as a target type a most encompassed type from among a first set of types of loosely-typed operands, where said first set of types comprises all resulting types of all first variant expressions, where each of said first variant expressions comprises said target expression with at least one of said expression operands, ~~wherein~~ at least one of said expression operands ~~is of an undefined~~ being a user-defined type, replaced by a widening type conversion, if said first set is not empty;

if said first set is empty, determining as said target type a most encompassing type from among a second set of types of loosely-typed operands, where said second set of types comprises all resulting types of all second variant expressions, where each of said second variant expressions comprises said target expression with at least one of said expression operands replaced using at least one of widening and narrowing type conversion; and

assigning said target type to said ~~undefined~~ operator.

11. (Previously presented) The computer-readable storage medium of claim 10, where said operands comprise n operands O_1 through O_n, where n is greater than or equal to one, and where each operand O_m is of a specific type T_m, where said act of determining as said target type said most encompassed type from among said first set of types comprises:

calculating said first set comprising types resulting from the operation of said operator on any possible set of hypothetical operands HO_1 through HO_n, where each hypothetical operand HO_m is of a type to which there is a widening conversion from type T_m ; and

if said first set is not empty, determining as said target type a most encompassed type from among said first set.

12. (Previously presented) The computer-readable storage medium of claim 11, wherein:

if said first set is empty, calculating a second set comprising types resulting from the operation of said operator on any possible set of hypothetical operands HO_{n+1} through HO_{n+n}, where each hypothetical operand HO_{n+m} is of a type to which there is a conversion, either narrowing or widening, from type T_m; and

determining as said target type a most encompassing type from among said second set.

13. (Previously presented) The computer-readable storage medium of claim 11, said acts further comprising:

converting each of said operands O_1 through O_n to said target type; and

computing said operation on said converted operands O_1 through O_n.

14. (Previously presented) The computer-readable storage medium of claim 11 where said operator is a binary operator and n equals two.

15. (Previously presented) The computer-readable storage medium of claim 11 where said operator is a unary operator an n equals one.

16. (Previously presented) The computer-readable storage medium of claim 10 where said target type is an intrinsic type.

17. (Currently Amended) A computer-readable storage medium containing computer executable instructions for a compiler to resolve an expression comprising an overloaded binary operator, a first operand of a first type and a second operand of a second type, in a loosely-typed programming language, the computer-executable instructions to perform acts comprising:

parsing the expression to determine the overloaded binary operator, the first operand and the second operand, the first operand and the second operand each being loosely typed, at least one of the first operand and the second operand being a user-defined type, ~~wherein at least one of the set comprising the first operand and the second operand is of an undefined type~~;

determining a first set of types, where said first set comprises all types to which there is a widening conversion from said first type;

determining a second set of types, where said second set comprises all types to which there is a widening conversion from said second type;

determining a third set of types, where said third set comprises all types which result from [[the]] an operation of said overloaded binary operator on a type from among said first set and a type from among said second set;

if said third set of types is empty, determining a fourth set of types, where said fourth set comprises all types to which there is a narrowing conversion from said first type and all types to which there is a widening conversion from said first type;

if said third set of types is empty, determining a fifth set of types, where said fifth set comprises all types to which there is a narrowing conversion from said second type and all types to which there is a widening conversion from said second type;

if said third set of types is empty, determining a sixth set of types, where said sixth set comprises all types which result from the operation of said overloaded binary operator on a type from among said fourth set and a type from among said fifth set;

if said third set of types is not empty, selecting the most encompassed type in said third set as a target type; and

if said third set of types is empty, selecting the most encompassing type in said sixth set of types as said target type; and

assigning said target type to said ~~undefined~~ overloaded binary operator.

18. (Currently Amended) The computer-readable storage medium of claim 17, said acts further comprising:

converting said first operand to said target type;

converting said second operand to said target type; and

computing said operation of said overloaded binary operator on said converted first operand and said converted second operand.

19. (Currently Amended) A system, comprising:

a microprocessor,

a computer readable memory, the memory containing instructions for determining a target type in an expression with an ~~undefined~~overloaded operator and at least one expression operand, further comprising:

determining as a target type a most encompassed type from among a first set of types of loosely-typed operands, where said first set of types comprises all resulting types of all first variant expressions, where each of said first variant expressions comprises said target expression with at least one of said expression operands, ~~wherein~~ at least one expression operand ~~is of an undefined~~ is a user-defined type, replaced by a widening type conversion, if said first set is not empty;

if said first set is empty, determining as said target type a most encompassing type from among a second set of types of loosely-typed operands, where said second set of types comprises all resulting types of all second variant expressions, where each of said second variant expressions comprises said target expression with at least one of said expression operands replaced using at least one of widening and narrowing type conversion; and

assigning said target type to said ~~undefined~~ overloaded operator.

20. (Currently Amended) The operator resolution system of claim 19, where said operands comprise n operands O_1 through O_n, where n is greater than or equal to one, and where each operand O_m is of a specific type T_m, where said first set determination module determining as a target type a most encompassed type from among a first set of types comprises:

calculating said first set comprising types resulting from the operation of said overloaded operator on any possible set of hypothetical operands HO_1 through HO_n, where each hypothetical operand HO_m is of a type to which there is a widening conversion from type T_m ; and

if said first set is not empty, determining as said target type a most encompassed type from among said first set.

21. (Currently Amended) The system of claim 20 wherein said memory further comprises instructions for:

if said first set is empty, calculating a second set comprising types resulting from the operation of said overloaded operator on any possible set of hypothetical operands HO_{n+1} through HO_{n+n}, where each hypothetical operand HO_{n+m} is of a type to which there is a conversion, either narrowing or widening, from type T_m; and

if said first set is empty, determining as said target type a most encompassing type from among said second set.

22. (Currently Amended) The system of claim 20, wherein said memory further comprises instructions for:

converting each of said operands O_1 through O_n to said target type; and

computing [[said]] an operation of said overloaded operator on said converted operands O_1 through O_n.

23. (Currently Amended) The system of claim 20 where said overloaded operator is a binary operator and n equals two.

24. (Currently Amended) The system of claim 20 where said overloaded operator is a unary operator and n equals one.

25. (Previously presented) The system of claim 19 where said target type is an intrinsic type.

26. (Currently Amended) A system, comprising:

a microprocessor,

a computer readable memory, the memory containing instructions for resolving an expression comprising an overloaded binary operator, a first operand of a first type and a second operand of a second type, the first operand and the second operand each being loosely-typed, further comprising:

determining a first set of types, where said first set comprises all types to which there is a widening conversion from said first type;

determining a second set of types, where said second set comprises all types to which there is a widening conversion from said second type;

determining a third set of types, where said third set comprises all types which result from [[the]] an operation of said overloaded binary operator on a type from among said first set and a type from among said second set;

if said third set of types is empty, determining a fourth set of types, where said fourth set comprises all types to which there is a narrowing conversion from said first type and all types to which there is a widening conversion from said first type;

if said third set of types is empty, determining a fifth set of types, where said fifth set comprises all types to which there is a narrowing conversion from said second type and all types to which there is a widening conversion from said second type;

if said third set of types is empty, determining a sixth set of types, where said sixth set comprises all types which result from the operation of said overloaded binary operator on a type from among said fourth set and a type from among said fifth set;

if said third set of types is not empty, selecting the most encompassed type in said third set as a target type; and

if said third set of types is empty, selecting the most encompassing type in said sixth set of types as said target type; and

assigning said target type to said ~~undefined~~overloaded binary operator.

27. (Currently Amended) The system of claim 26, wherein said memory further comprises instructions for:

converting said first operand to said target type;

converting said second operand to said target type; and

computing said operation of said overloaded binary operator on said converted first operand and said converted second operand.